Procesador: Problemas

Ejercicio 1

Determine el mínimo tiempo del ciclo de reloj del computador, para cada una de las siguientes posibilidades de su Unidad de Control:

a) cableada.

Una UC cableada dura tanto como la mayor duración de la operación elemental más larga, en este caso, el periodo mínimo es de 25 ut.

b) microprogramada.

Tmin= Tiempo lectura de registro + Tiempo de acceso a MC + Tiempo de escritura en el registro de control + tiempo de lectura en el RC + Tiempo operación más larga + Tiempo de lect. del RE + Tiempo de secuenciador + Tiempo de escritura de registro de µdir = (1+20+1+1+25+1+6+1)ut = 56 ut.

Ejercicio 2

Sea un computador con Unidad de Control µprogramada cuyo tiempo de ciclo es de 10 ut. La memoria es asíncrona y activa una señal READY cuando acaba el acceso que se le ha solicitado. Microprograme a nivel RT la instrucción de una palabra POP [--.Ri], indicando qué microinstrucciones pertenecen a cada una de las fases de ejecución de la instrucción incluyendo el fetch. Suponga que el puntero de pila crece en direcciones decrecientes de memoria y apunta al primer hueco en pila. Indique el tiempo medio de ejecución de dicha instrucción, suponiendo que el número medio de ciclos necesario para cada acceso a memoria es de 3 ciclos.

POP [--.Ri] → Extraer de la pila el último valor de la misma a la dirección del registro Ri predecrementandolo. Es decir, si Ri=3 se lleva a la dirección 2.

Fetch

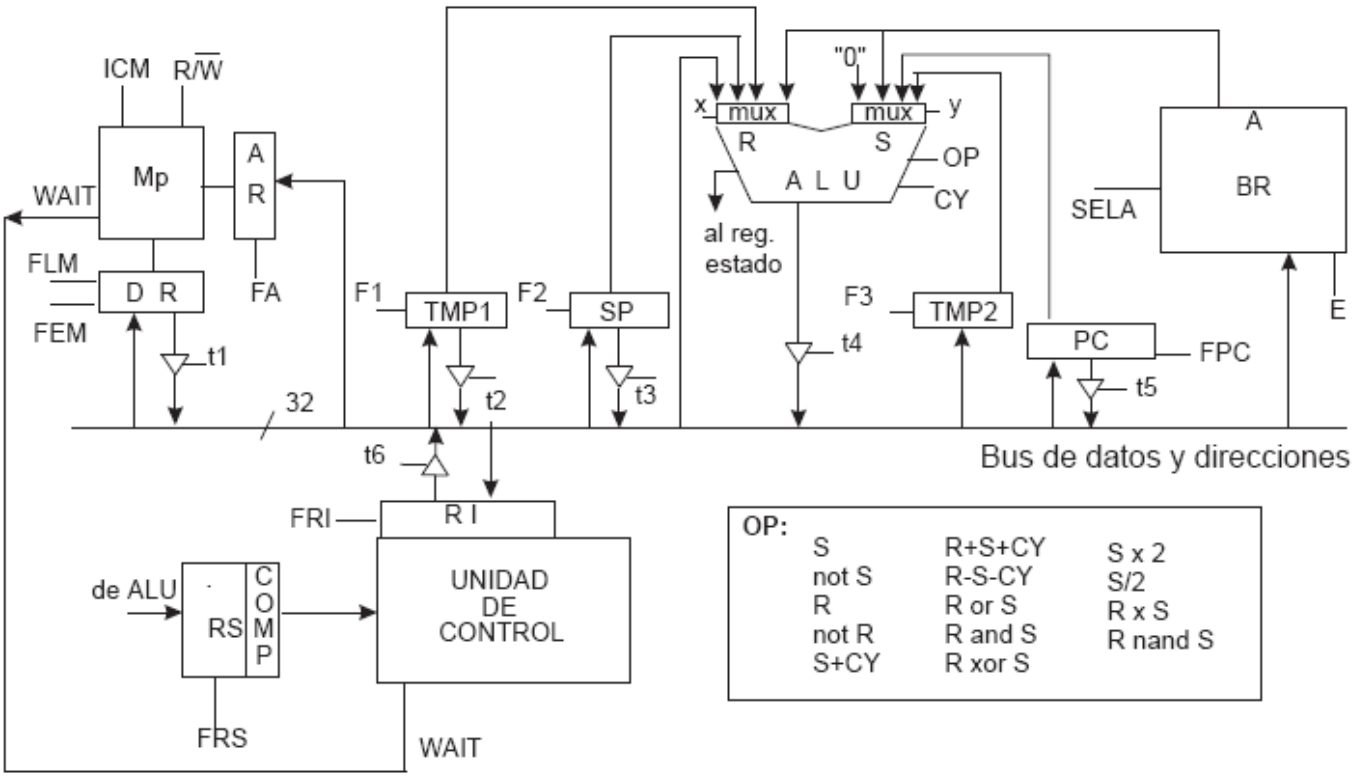
f1: AR ← PC
f2: DR ←M(AR)
si no READY µsalto a f2 } 3 ciclos
f3: IR ←DR
f4: PC ←PC++
µsalto a C.O.

Fase de ejecución

e1: SP ←SP++ ; Incremento el puntero de pila
e2: AR ← SP

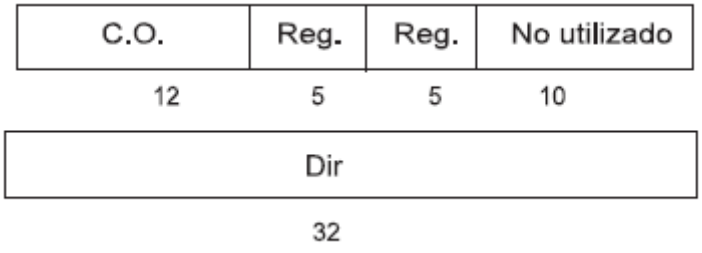
e3: DR ← M(AR)
si no READY µsalto a e3
e4: Ri ←Ri--
e5: AR ←Ri
e6: M(AR) ←DR
si no READY µsalto a e6
} 3 ciclos
e7: µsalto a fetch

Ejercicio 9



Se recomienda leer el ejercicio con atención, lentamente, reconociendo cada aspecto importante.

1. Realice, a nivel RT, el microprograma de la instrucción de dos palabras ADD INT .R1, .R2, /Dir cuyo formato se muestra en la figura 2:



Como siempre primero efectuamos el Fetch

f1: AR ←PC
f2: AR ← M(AR)
si WAIT µsalto a f2
f3: R1 ← AR
f4: PC ← PC + 1
µsalto a C.O.
5: TMP2 ← R1
6: TMP1 ← TMP2 + R2
Actualizar R5
7: Si Z µsalto a 16
8: AR ← PC
9: DR ← M(AR)
si WAIT µsalto a 9
10: AR ← DR
11: DR ← M(AR)
si WAIT µsalto a 11
12: R1 ← DR
13: R2 ← DR
14: DR ← TMP1
15: M(AR) ← DR
si WAIT µsalto a 15
16: PC ← Pc + 1
µsalto a fetch

2.-
200 instrucciones
30 microinstrucciones / µprograma
144 " para subrutinas

Número mínimo de palabras (µI's) de la MC

(200 instrucciones x 30 µI / µprograma (instrucción)) + 144 µI = 6144 µI = 6 K

Formato de µI

Memoria		Registros		Bus		BR		Secuencia.		Condiciones		ALU		µdirección	
0	4	5	10	11	13	14	19	20	21	22	25	26	34	35	47

Memoria

ICM	bit 0
R/W	bit 1
Carga AR	bit 2
Carga DR	bits 3, 4

Registros

6 regs	bits 5 a 10

Bus

6 triestados	bits 11 a 13

BR

SELA	bits 14 a 18
E	bit 19

Secuenciamento

bits 20, 21

Condiciones µsalto

12 condiciones	bits 22 a 25

ALU

x	bits 26, 27
y	bits 28, 29
CY	bit 30
OP	bits 31 a 34

Ejercicio 11

Dados los tiempos

$t_{lect/escr}$ BR = 3ns
t_{op} más lento de la ALU = 10 ns
$t_{triestado}$ = 1 ns
$t_{lect/escr\ regs}$ = 2 ns
t_{MC} = 20 ns
t_{sec} = 3 ns

¿Cuál es el T_{CK}?

T_{CK} = t_{op} elemental de mayor duración (camino crítico) + t_{UC} µprogramada

$t_{camino\ crítico} = t_{BR} + t_{ALU} + t_{triestado} + t_{BR}$ = (3 + 10 + 1 + 3) ns = 17 ns

t_{UC} µprogramada = t_{lect} RS + t_{sec} + t_{MC} = (2 + 3 + 20) = 25 ns

T_{CK} =(17 + 25) ns = 42 ns

segunda parte

CALL_ADDC .R4, [.R5++], $ desp

R4 ← R4 + [R5]
Si acarreo: desplazamiento desp

<u>subrutina de fetch</u>

1: AR ← PC
2: DR ← M(AR)
Si WAIT µsalto a 2
3: RI ← DR
4: PC ← PC + 1, µsalto a CO
5: AR ←R5
6: DR ← M(AR)
Si WAIT µsalto a 6
7: TMP2 ← DR
8: R4 ← TMP2 + R4
Actualizar RS
9: R5 ← R5 + 1 ------> *¿Si hay una suma por qué no se actualiza el RS?* Porque es un incremento de registro y el RS solo se actualiza en las operaciones aritméticas o lógicas de operandos no en decrementos e incrementos de registros.
Si NC µsalto a fetch
10: AR, SP ← SP +1
11: DR ← PC
12: M(AR) ← DR
Si WAIT µsalto a 12
13: TMP1 ← RI(desp)
14: PC ← TMP1 + PC
µsalto a fetch.

Ejercicio de examen

Unidad de Control microprogramada con cuatro registros temporales y con un µcontador para µbucles. La señal de reloj es asíncrona. La Mp es direccionable a nivel de byte. Se quiere dotar a este computador de instrucciones múltiples que operan con operandos vectoriales. Entre ellas está la instrucción de una palabra:

PUSHM #n, [.Ri]

Esta instrucción introduce en pila n palabras apuntadas por el registro Ri cuyo valor no queda modificado tras la instrucción. Sp apunta al último dato introducido en la pila y la pila crece hacia direcciones decrecientes de memoria.

El fetch es como siempre, así que ahora lo obviamos.

m1: TMP1 ← Ri
µcontador ← n
m2: decrementa µcontador
si µcontador = 0 µsalto a fetch.
m3: AR ← TMP1
m4: DR ← M(AR)
Si no Ready µsalto a m4
m5: AR, SP ← SP – 4
m6: M(AR) ← DR
Si no Ready µsalto a m6
m7: TMP1 ← TMP1 + 4
decrementar µcontador
si µcontador ≠ 0
ir a m3
m8: µsalto a fetch